Apache HTTP Server

Configuration Files

Main Configuration Files

Related Modules	Related Directives
mod_mime	AccessConfig <IfDefine> Include ResourceConfig TypesConfig

Apache is configured by placing directives in plain text configuration files. The main configuration file is usually called `httpd.conf`. The location of this file is set at compile-time, but may be overridden with the `-f` command line flag. Some sites also have `srm.conf` and `access.conf` files for historical reasons. In addition, other configuration files may be added using the `Include` directive. Any directive may be placed in any of these configuration files. Changes to the main configuration files are only recognized by Apache when it is started or restarted.

New with Apache 1.3.13 is a feature where if any configuration file is actually a directory, Apache will enter that directory and parse any files (and subdirectories) found there as configuration files. One possible use for this would be to add VirtualHosts by creating small configuration files for each host, and placing them in such a configuration directory. Thus, you can add or remove VirtualHosts without editing any files at all, simply adding or deleting them. This makes automating such processes much easier.

The server also reads a file containing mime document types; the filename is set by the TypesConfig directive, and is `mime.types` by default.

Syntax of the Configuration Files

Apache configuration files contain one directive per line. The back-slash "\" may be used as the last character on a line to indicate that the directive continues onto the next line. There must be no other characters or white space between the back-slash and the end of the line.

Directives in the configuration files are case-insensitive, but arguments to directives are often case sensitive. Lines which begin with the hash character "#" are considered comments, and are ignored. Comments may **not** be included on a line after a configuration directive. Blank lines and white space occurring before a directive are ignored, so you may indent directives for clarity.

You can check your configuration files for syntax errors without starting the server by using `apachectl configtest` or the `-t` command line option.

Modules

Related Modules	Related Directives
mod_so	AddModule ClearModuleList <IfModule> LoadModule

Apache is a modular server. This implies that only the most basic functionality is included in the core server. Extended features are available through modules which can be loaded into Apache. By default, a base set of modules is included in the server at compile-time. If the server is compiled to use dynamically loaded modules, then modules can be compiled separately and added at any time using the LoadModule directive. Otherwise, Apache must be recompiled to add or remove modules. Configuration directives may be included conditional on a presence of a particular module by enclosing them in an <IfModule> block.

To see which modules are currently compiled into the server, you can use the `-l` command line option.

Scope of Directives

Related Directives
<Directory> <DirectoryMatch> <Files> <FilesMatch> <Location> <LocationMatch> <VirtualHost>

Directives placed in the main configuration files apply to the entire server. If you wish to change the configuration for only a part of the server, you can scope your directives by placing them in `<Directory>`, `<DirectoryMatch>`, `<Files>`, `<FilesMatch>`, `<Location>`, and `<LocationMatch>` sections. These sections limit the application of the directives which they enclose to particular filesystem locations or URLs. They can also be nested, allowing for very fine grained configuration.

Apache has the capability to serve many different websites simultaneously. This is called Virtual Hosting. Directives can also be scoped by placing them inside `<VirtualHost>` sections, so that they will only apply to requests for a particular website.

Although most directives can be placed in any of these sections, some directives do not make sense in some contexts. For example, directives controlling process creation can only be placed in the main server context. To find which directives can be placed in which sections, check the Context of the directive. For further information, we provide details on How Directory, Location and Files sections work.

.htaccess Files

Related Directives
AccessFileName AllowOverride

Apache allows for decentralized management of configuration via special files placed inside the web tree. The special files are usually called `.htaccess`, but any name can be specified in the `AccessFileName` directive. Directives placed in `.htaccess` files apply to the directory where you place the file, and all sub-directories. The `.htaccess` files follow the same syntax as the main configuration files. Since `.htaccess` files are read on every request, changes made in these files take immediate effect.

To find which directives can be placed in `.htaccess` files, check the Context of the directive. The server administrator further controls what directives may be placed in `.htaccess` files by configuring the `AllowOverride` directive in the main configuration files.

For more information on `.htaccess` files, see Ken Coar's tutorial on Using .htaccess Files with Apache.

Apache HTTP Server